

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims

1. (Currently Amended) A display system, comprising:

a display;

a graphical user interface;

a processor coupled between the display and the graphical user interface and adapted to operate from a windows-based operating system for executing a software component that, during runtime of an application program, wherein the processor-executed software component generates a first image upon the display independent of code within the operating system during a first time and, during a second time, emulates code that, when executed by the processor, generates a second image upon the display dependent on code within the operating system, and wherein the first and second images are substantially identical.

2. (Original) The display system as recited in claim 1, wherein said first and second images have the same look and feel.

3. (Original) The display system as recited in claim 2, wherein the first and second images comprise pixels presented upon the display via the graphical user interface associated with the application program.

4. (Currently Amended) The display system as recited in claim 4~~2~~, wherein the first and second images comprise images of an object ~~is selected~~ from a group comprising buttons, list boxes and slide bars on which a pointer device can be directed by a user.

5. (Original) The display system as recited in claim 1, wherein the application program is written in Java programming language.

6. (Original) The display system as recited in claim 5, wherein said software component comprises a Java application program interface consisting of an abstract windowing toolkit (AWT) during the second time.

7. (Original) The display system as recited in claim 5, wherein said software component comprises a Java application program interface consisting of a Swing application program interface during the first time.

8. (Currently Amended) The display system as recited in claim 1, wherein the operating system comprises a Windows, Unix or OS/2 computer operating system.

9. (Currently Amended) The display system as recited in claim 1, wherein the first and second images present the same look and feel upon the display independent of the operating system.

10. (Currently Amended) A method for displaying an image, comprising:

running an application program upon a computer and under an operating system, wherein the application program includes a first interface dependent on the operating system;

displaying a first image upon a display of the computer using the first interface;

replacing the interface with a second interface that is substantially independent of the operating system yet emulates the behavior of at least a part of the first interface;

re-running the application program; and

re-displaying a second image upon the display of the computer using the second interface, wherein the second image has ~~the~~ substantially the same look and feel as the first image.

11. (Original) The method as recited in claim 10, wherein said displaying and re-displaying comprises presenting pixels upon the display via a graphical user interface associated with the application program.

12. (Currently Amended) The method as recited in claim 10, wherein ~~the said~~ displaying and re-displaying comprises presenting objects selected from a group comprising buttons, list boxes and slide bars upon the display on which a pointer device can be directed by a user.

13. (Original) The method as recited in claim 10, wherein the application program is written in Java programming language.

14. (Original) The method as recited in claim 10, wherein said running comprises implementing the first interface as a Java application program interface consisting of an abstract windowing toolkit (AWT).

15. (Currently Amended) The ~~system-method~~ as recited in claim 10, wherein said re-running comprises implementing the second interface as a Java application program interface consisting of a Swing application program interface that draws the second image over any other image that exists within the area of the display adapted to receive the second image.

16. (Original) The method as recited in claim 10, wherein said running and re-running comprises operating the computer on a Windows, Unix or OS/2 operating system.

17. (Original) The method as recited in claim 10, wherein the look and feel of the second object is independent of the operating system.

18. (Currently Amended) A computer-readable storage device, comprising:

an operating system; and

an application program adapted for executing code of a software component, which:

during a first time, ~~to generate~~ generates a first image ~~independent~~ dependent of ~~executing~~ code executing within the operating system; and

during a second time, ~~to generate~~ generates a second image ~~dependent~~ independent on ~~executing code~~ executing within the operating system, wherein the first-second image is adapted to overwrite ~~an the first image~~ upon a display screen ~~previous to the first image during the second time, yet and wherein the first image cannot overwrite the and second image images upon the display screen during the first time~~ are substantially identical.

19. (Currently Amended) The computer-readable storage device as recited in claim 18, further comprising an application interface code that executes, during the first time, separate and apart from the code within the operating system, and wherein said application interface code is written in a Java programming language as a Swing application program.

20. (Currently Amended) The computer-readable storage device as recited in claim 18, further comprising an application interface code that executes, during the second time, from code within the operating system, and wherein said application interface code is written in a Java programming language as an abstract windowing toolkit (AWT).